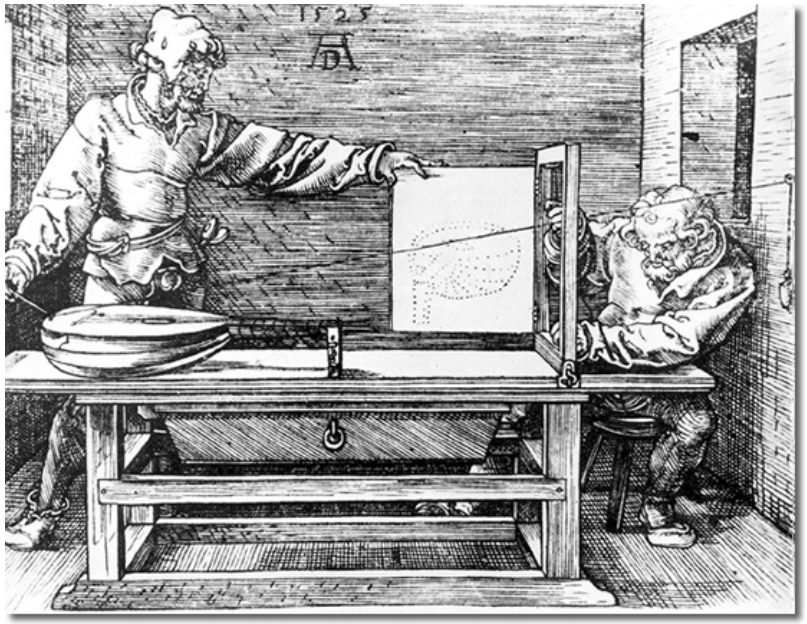
Transformations 3D->2D
Perspective

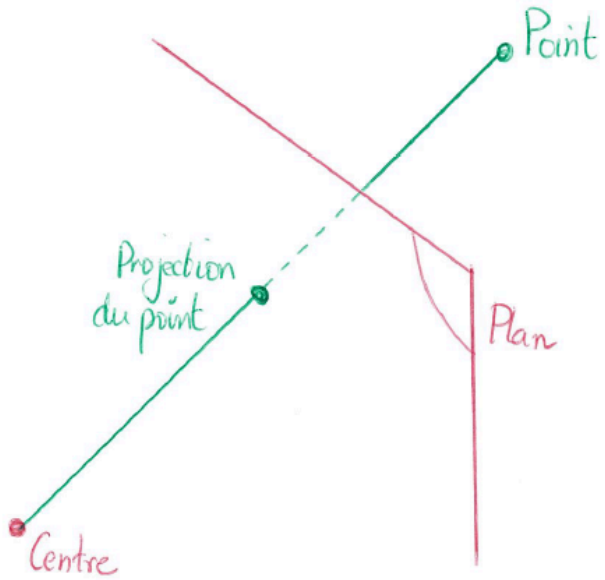
Perspective

Projection Centrale

- Transformation de 3D vers 2D
  - Un point (centre)
  - Un plan (écran)

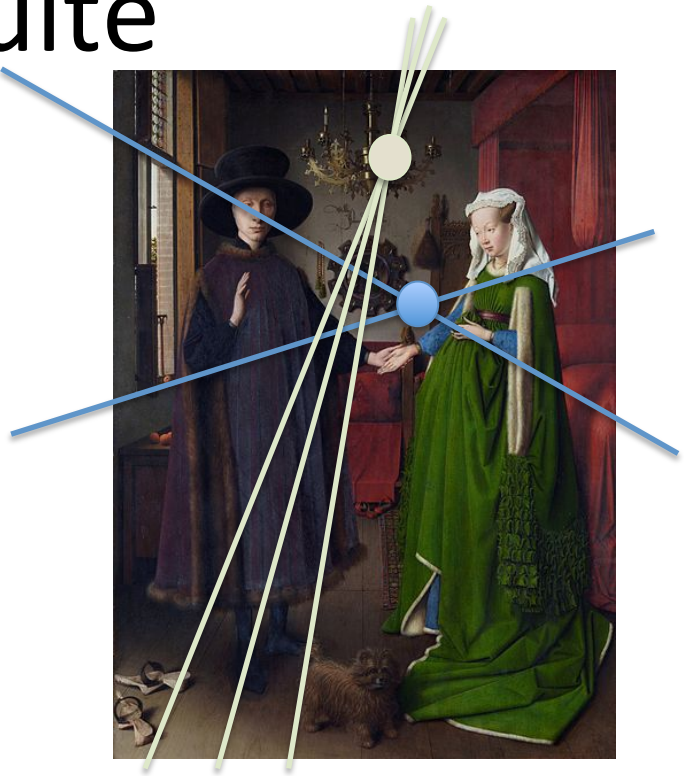


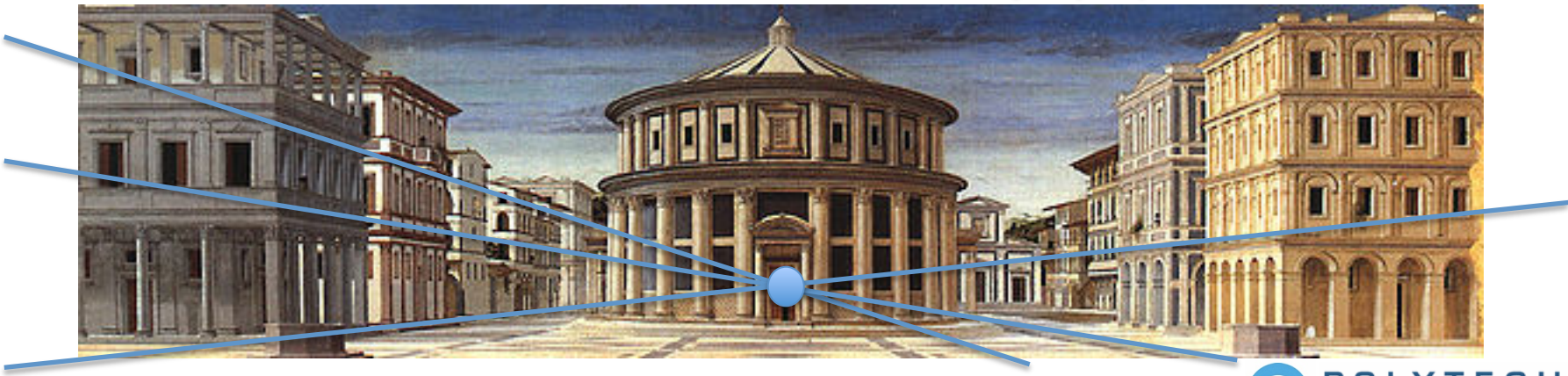
Albert Dürer (fin XV)



- Une droite se projette en une droite
- 2 droites // se projettent en deux droites concourantes
  - ➢ Intersection: point de fuite

Points de fuite





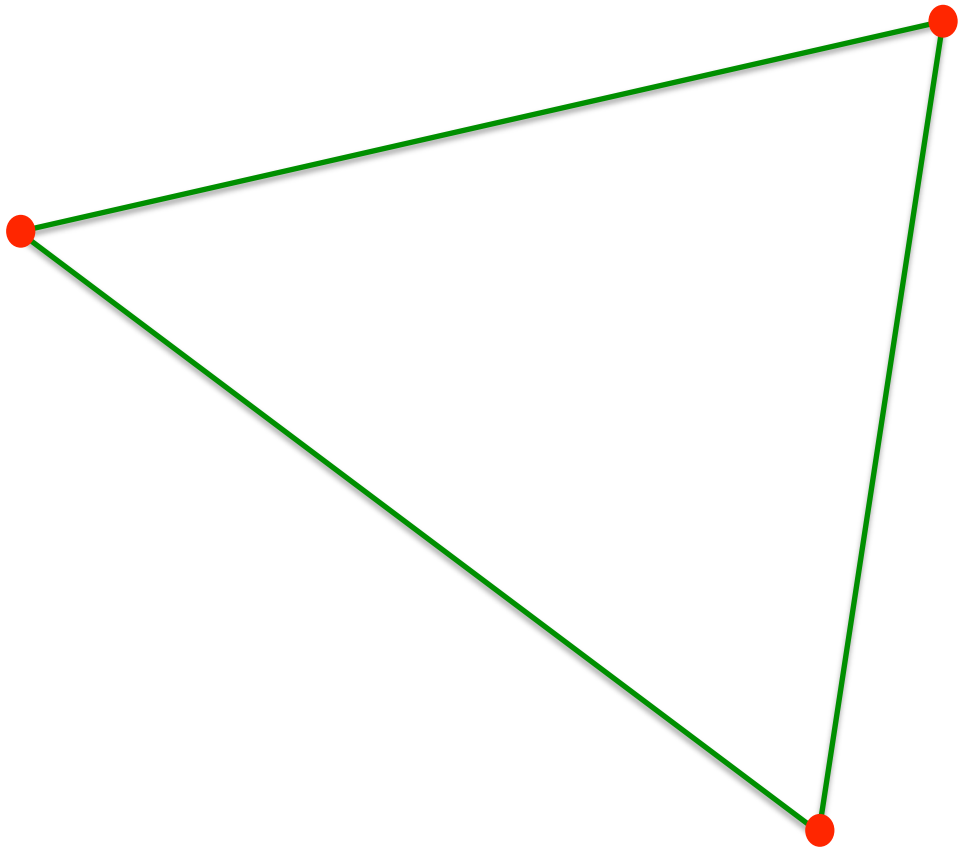
Dessiner les points de fuite

G.-F

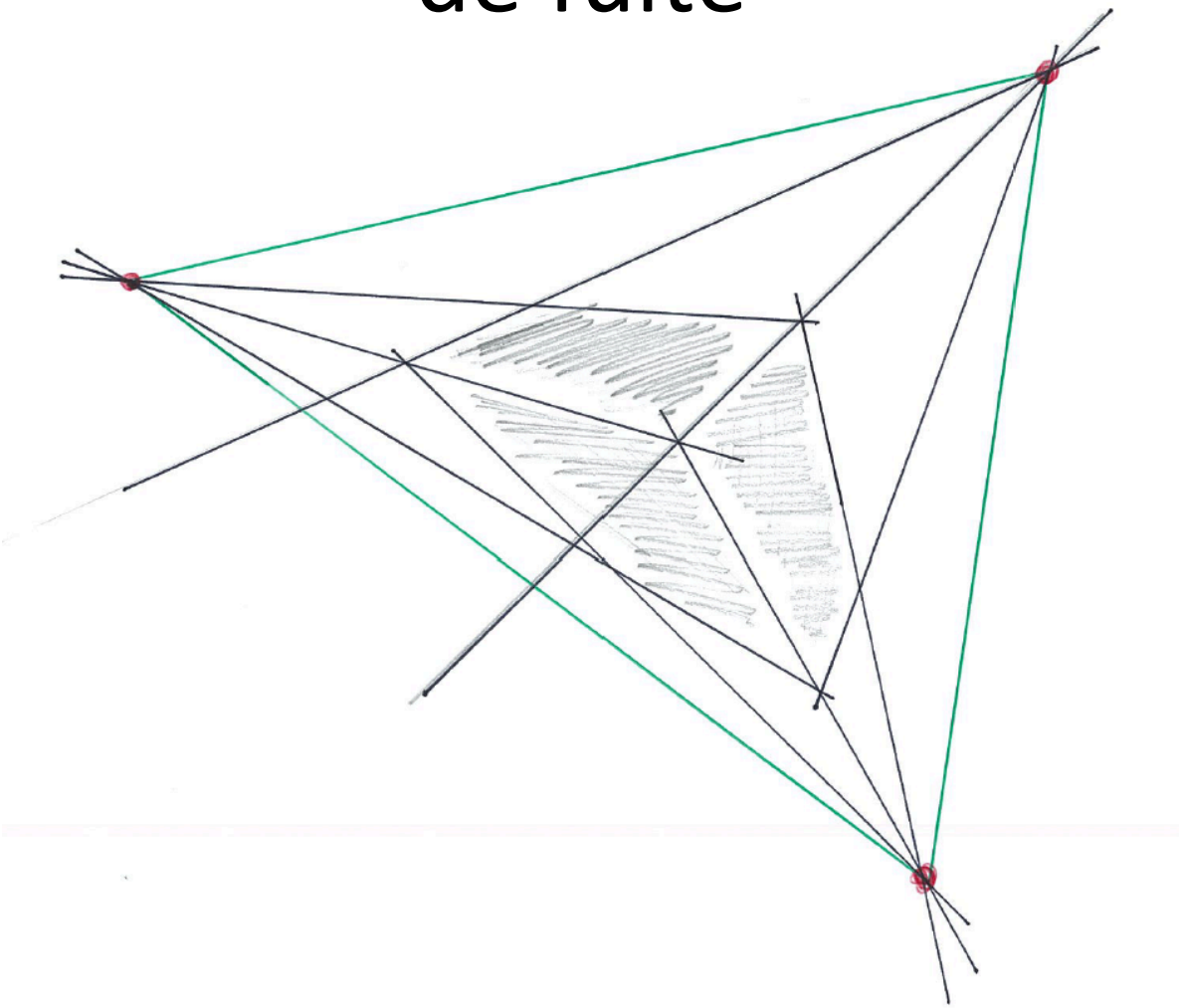
CH

- Prendre une photo d'une caisse, avec des points de fuite le plus proche possible des bords de la photo
  - Oblige à « penser » 3D et 2D *en même temps*
  - Demo avec Blender !!!

Dessiner un cube connaissant 3 points de fuite



Dessiner un pavé connaissant 3 points de fuite



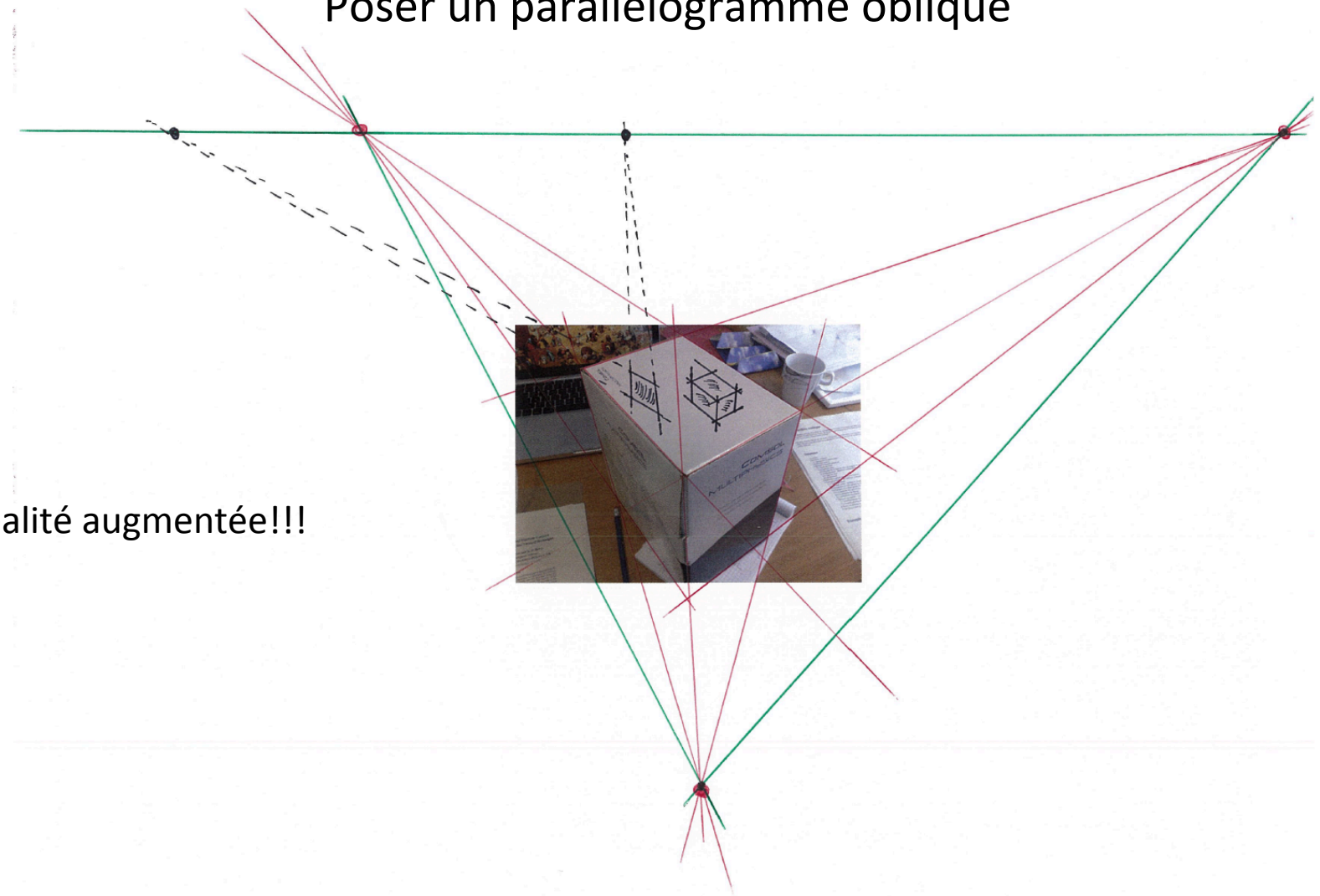
Empiler un pavé

Poser un parallélogramme oblique

G.-P. Bonneau

Empiler un cube

Poser un parallélogramme oblique

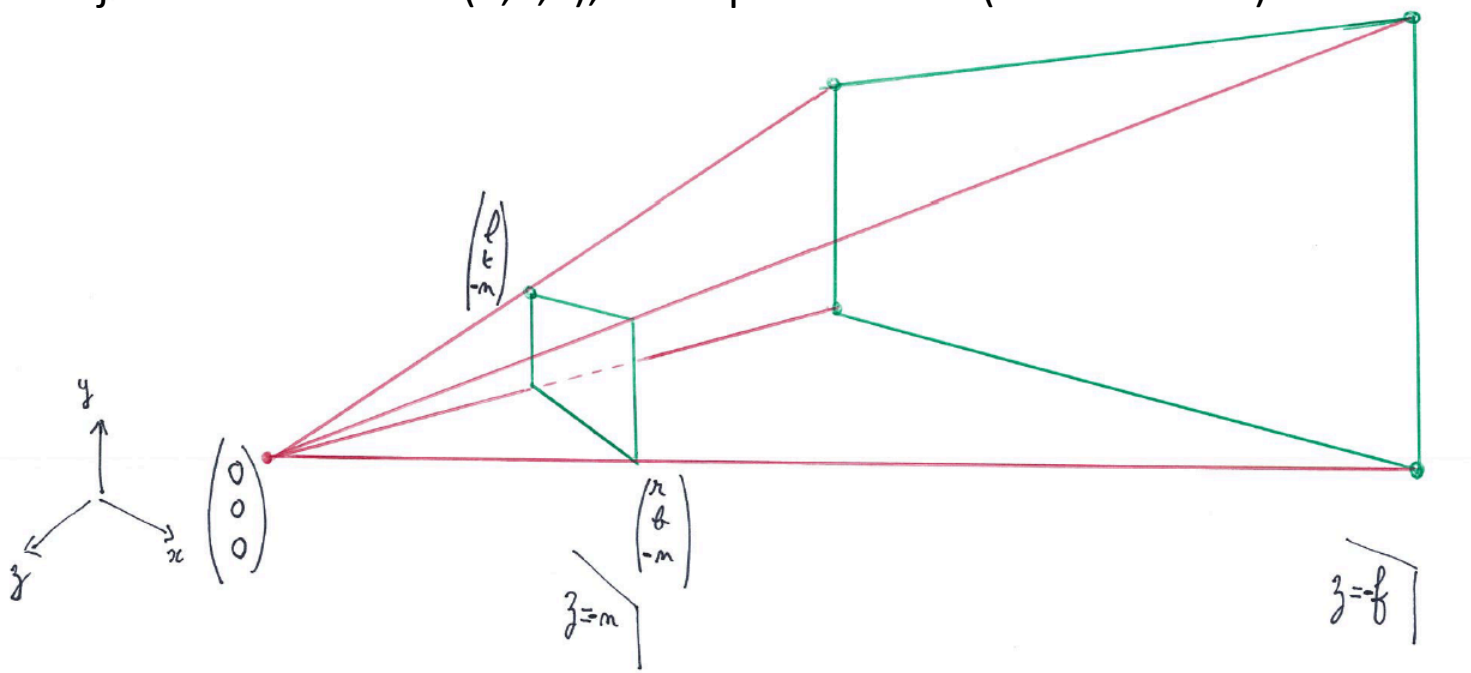


Réalité augmentée!!!

Perspective en OpenGL

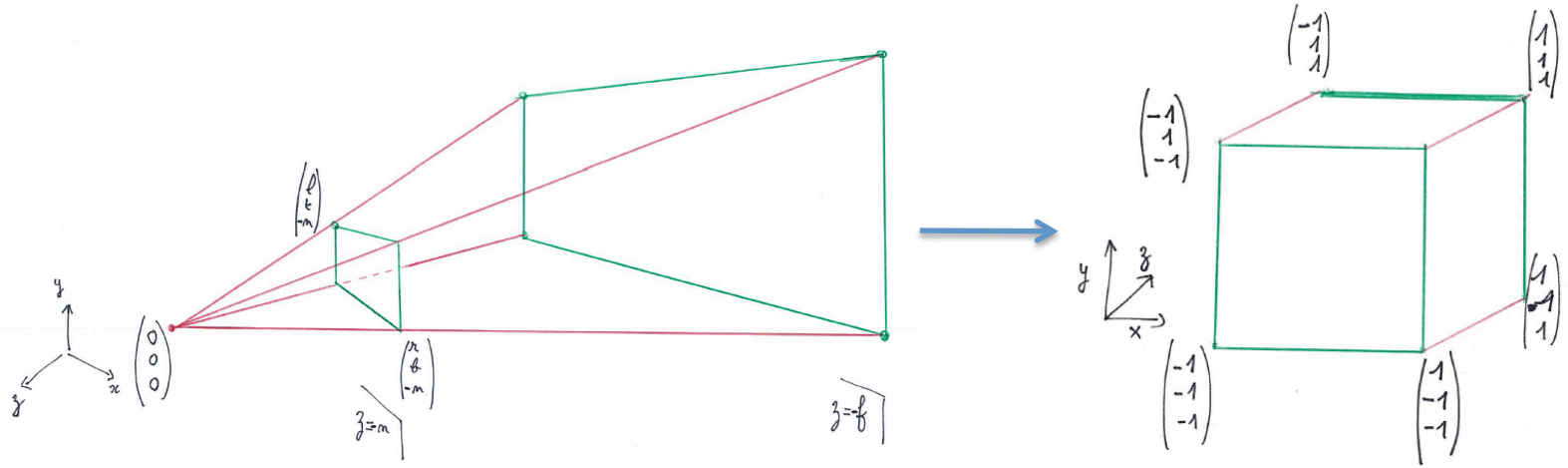
```
glm::frustum(left, right, bottom, top, near, far);
```

- « Frustum »: pyramide tronquée
- 6 flottants pour définir 2 rectangles verticaux
- Projection centrée en (0,0,0), sur le plan z=-near (0 < near < far)



Perspective en OpenGL

- 3D -> 2D… + profondeur!



$$\begin{pmatrix} x_e \\ y_e \\ z_e \\ w_e \end{pmatrix}$$

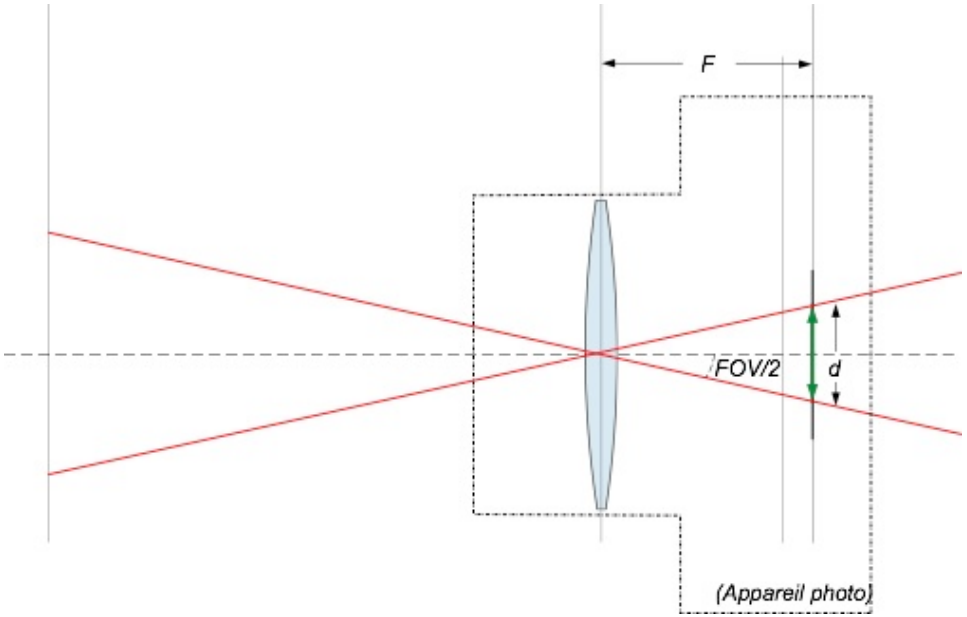
$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ w_c \end{pmatrix}$$

Matrice de projection

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ w_c \end{pmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2nf}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \\ w_e \end{pmatrix}$$

Coordonnées homogènes: $\begin{pmatrix} aw \\ bw \\ cw \\ w \end{pmatrix}$ et $\begin{pmatrix} a \\ b \\ c \\ 1 \end{pmatrix}$ correspondent au même point 3D

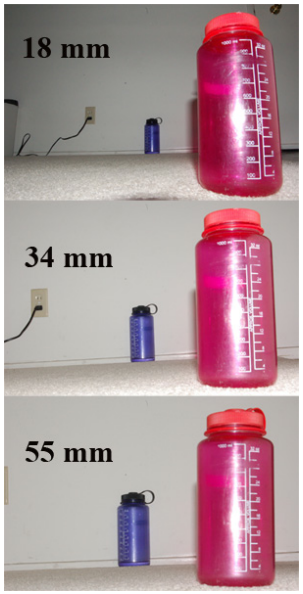
Champ visuel



d: taille du film
F: distance focale
FOV: angle de vue (field of view)

$$\tan(\frac{\mathrm{FOV}}{2}) = \frac{(d/2)}{F}$$

Aussi une projection centrale!!!



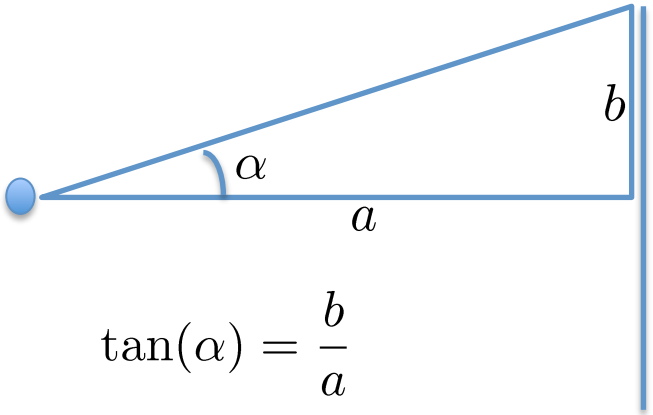
FOV grand
Appareil proche

FOV petit
Appareil éloigné

Pyramide de vue vs. angle de vue

- glm:frustum(l, r, b, t, n, f)
- glm::perspective(FOV, aspect, near, far)
  - FOV: angle de vue, en degré
  - aspect: rapport largeur/hauteur fenêtre

- Equivalence?



$$\tan(\alpha) = \frac{b}{a}$$